S2-63

3P

188181

Collision-Free Trajectory Planning Algorithm for Manipulators

F. Pourboghrat and J.Y. Han
Southern Illinois University
Carbondale, IL 62901

1. ~~Abstract~~

Collision-free trajectory planning for robotic manipulators is investigated, ~~in this paper.~~ The task of the manipulator is to move its end-effector from one point to another point in an environment with polyhedral obstacles. An on-line algorithm is developed based on finding the required joint angles of the manipulator, according to goals with different priorities. The highest priority is to avoid collisions, the second priority is to plan the shortest path for the end effector, and the lowest priority is to minimize the joint velocity for smooth motion. The pseudo-inverse of the Jacobian matrix is applied for inverse kinematics. When a possible collision is detected, a constrained inverse kinematic problem is solved such that the collision is avoided. This algorithm can also be applied to a time-variant environment.

2. Introduction

Ordinary tasks for a robotic manipulator are to move its end effector from an admissible point to another admissible point in an environment with obstacles. For that, the initial and final configuration of the manipulator are often given for the trajectory planning. Usually, there are infinite paths for the end effector. Even for a specific path of the end effector, there are still infinite trajectories possible for the manipulators. However, some of the trajectories are not feasible becasue of arm geometry, obstacles, and some kinematic or dynamic constraints. Even with the kinematically feasible trajectories, some computational or logic problems in the algorithms may make them impractical.

3. Trajectory Planning

In order to move from one point to another, in the task space, one needs to solve for the angular information from the spatial information, using the tnverse kinematic relationship. Consider a robotic manipulator with n degrees of freedom. Let the kinematic relationship between joint angles and the end-effector position and orientation be given by

$$X = f(q) \tag{1}$$

where X is the m-dimentional vector of the end-effector position and orientation, and q is the n-dimensional vector of joint angles. For a kinematically redundant manipulator, the dimension of q is greater than the dimension of X, (n>m). Differentiating the above relation, we get

$$\dot{X} = J(q)\dot{q} \tag{2}$$

where $J(q) = df/dq$ is the mxn Jacobian matrix, [7]. For a redundant manipulator, the Jacobian matrix will have more columns than rows. Moreover, the inverse of such non-square matrix is not defined in a regular sense. However, useful solutions of equation (2) can be found, by using the generalized-inverses of the Jacobian matrix J, and is given by

$$\dot{q} = J^{\dagger}\dot{X} + (I-J^{\dagger}J)Z \tag{3}$$

where $J^{\dagger} = J^{T}(JJ^{T})^{-1}$ is the pseudo-inverse of J, I is the nxn identity matrix, and Z is an arbitrary n-dimensional vector. When the vector Z is selected to be zero, equation (3) reduces to

$$\dot{q} = J^{\dagger}\dot{X} \tag{4}$$

which gives the best approximate solution to equation (2). This is in the sense that if $\dot{q}_*$ is the solution of (2), given by (4), then $||\dot{q}_*|| < ||\dot{q}||$, where $\dot{q}$ is any other solution of (2) that is given by (3), [5-6]. It should be noted that, such minimum norm, or best approximate solution, is defined when there is no restriction in the task space. This means that, in a restricted environment, the above mentioned best approximate, solution may not be feasible for application and may result in collision.

PAGE 124 INTENTIONALLY BLANK

Let us define collision point to be the point on the manipulator body which has the potential to collide with the obstacle. The collision-free trajectory planning problem here is to develop an algorithm, for the on-line determination of the required joint angle rates, $\dot{q}$, for safe manipulator motion. The approach is to continuously monitor the task space, for detecting possible collisions. If no potential collision is detected, the required joint angle rates are generated, using the best approximate solution, to move the end-effector on a shortest distance. But when a potential collision point is detected, the trajectory is modified in order to avoid collision.

4. Obstacle Avoidance

In order to avoid obstacles, one needs to use the kinematic relationship for the collision points, similar to that of equations (1) and (2). Let the potential collision point, in the task space, be denoted by X_c. Then, similar to equation (2), we can write

$$\dot{X}_c = J_c(q)\dot{q} \tag{5}$$

where J_c is the mxn Jacobian matrix for the collision point. The inverse kinematic solution to the above is similar to (3) and is given by

$$\dot{q} = J_c^{\dagger}\dot{X}_c + (I-J_c^{\dagger}J_c)Z' \tag{6}$$

where $J_c^{\dagger}$ is the pseudo-inverse of J_c, and Z' is an arbitrary n-dimensional vector.

Now, the problem of obstacle avoidance is that, when a potential collision is detected, the highest priority is to avoid the obstacle, and,if needed modify the position of end-effector. In order for the trajectory planning to have minimum norm, we choose $Z = 0$ as in (4). On the other hand we choose $Z' \neq 0$, like in (6), to account for both collision avoidance and trajectory planning. From (3) and (6), a minimum norm solution for Z' is

$$Z' = [J(I-J_c^{\dagger}J_c)]^{\dagger}[\dot{X}-JJ_c^{\dagger}\dot{X}_c].$$

Plugging this back into (6), we get

$$\dot{q} = J_c^{\dagger}\dot{X}_c + (I-J_c^{\dagger}J_c)[J(I-J_c^{\dagger}J_c)]^{\dagger}[\dot{X}-JJ_c^{\dagger}\dot{X}_c].$$

Then, using the following identity, [6]

$$(I-J_c^{\dagger}J_c)[J(I-J_c^{\dagger})]^{\dagger} = [J(I-J_c^{\dagger}J_c)]^{\dagger}$$

we get

$$\dot{q} = J_c^{\dagger}\dot{X}_c + [J(I-J_c^{\dagger}J_c)]^{\dagger}[\dot{X}-JJ_c^{\dagger}\dot{X}_c]. \tag{7}$$

The above relation, generates the joint angle rates q such that the obstacle is avoided and the end effector velocity is modified, for the on-line trajectory planning.

Now the question is how and in what direction the end effector spatial velocity should be changed. For the algorithm to be fast and implementable, a finite search for the minimum norm solution is considered. The value of $\dot{X}_c$ is preselected by the user, and the value of $\dot{X}$ modification is also preselected by a value of ϵ. The value of $\dot{q}$ may now be found by examining seven different directions for rate modification, e.g., ϵ-variation in plus or minus X,Y,Z coordinates and also no modification. The smallest norm $||\dot{q}||$ is then chosen for trajectory modification, from seven different possibilities.

The overall algorithm is such that when no potential collision is detected, a minimum norm solution $\dot{q}$ is planned according to (4). But, when a potential collision is detected, the obstacle is avoided and the path in modified according to (7).

In order to develop the algorithm, it is assumed that:

(1) the solution exists
(2) the obstacles are represented by polyhedrals.
(3) the geometrical information about the task area is known, e.g., using sensory systems, the positions of obstacles and manipulator links are known.
(4) the potential collision point on the manipulator links can be detected.
(5) only a single collision may occur, and will be detected, at any given time.

5. The Algorithm

The following summarizes the steps, involved in the proposed algorithm, for the on-line collision-free trajectory planning of the robotic manipulators. The steps of the algorithm are:

(1) Determine a minimum-length path for the end-effector, from the current position to target position.

(2) Check if there is a potential collision point. If there is, go to step (5), otherwise continue.

(3) No potential collision is detected. Make an incremental move according to the joint angle rates vector $\dot{q}$, given by equation (4).

(4) If the end-effector has not reached the target, go to step (2). If it has reached the target, go to step (7).

(5) Potential collision is detected. Make an incremental move according to the joint angle rates vector $\dot{q}$, given by equation (7), such that $||\dot{q}||$ is minimized in a finite search.

(6) Go to step (1).

(7) Stop.

6. Conclusion

On-line, collision-free trajectory planning is discussed. An algorithm, which utilizes sensed information about the configuration of the manipulator and obstacles, is developed based on the task priorities. The order of the task priorities are: to avoid collision, to plan the shortest path for the end effector, and to choose the minimum norm solution. The algorithm is fast and could be implemented on robotic manipulators for on-line trajectory planning.

REFERENCES

[1] Lozano-Perez, T. and Wesley, M.A., "Algorithm for Planning Collision-Free Paths Among Polyhedral Obstacles," Comm. of ACM, Vol. 22, No. 10, Oct. 1979.

[2] Gouzenes, L., "Strategies for Solving Collision-Free Trajectories Problems for Mobile and Manipulator Robots," Int. J. Robotics Research, Vol. 3, No. 4, 1984.

[3] Khatib, O., "Real-Time Obstacle Avoidance for Manipulators and Mobile Robots," Int. J. Robotics Research, Vol. 5, No. 1, 1986.

[4] Gilbert, E.G. and Johnson, D.W., "Distance Functions and their Application to Robot Path Planning in the Presence of Obstacles," IEEE J. Robotics and Automation, Vol. RA-1, No. 1, March 1985.

[5] Klein, C.A. and Huang, C.H., "Review of Pseudoinverse Control for Use with Kinematically Redundant Manipulators," IEEE Trans. Syst. Man Cyb., Vol. SMC-13, No. 3, March/April 1983.

[6] Majciejewski, A.A. and Klein, C.A., "Obstacle Avoidance for Kinematically Redundant Manipulators in Dynamically Varying Environments," Int. J. Robotics Research, Vol. 4, No. 3, 1985.

[7] Lee, C.S.G., et. al., Tutorial on Robotics, IEEE Computer Society, 1983.